

# Spectral Clustering with Jensen-type kernels and their multi-point extensions

Debarghya Ghoshdastidar, Ambedkar Dukkipati, Ajay P. Adsul and Aparna S. Vijayan  
 Department of Computer Science & Automation  
 Indian Institute of Science  
 Bangalore 560012, India

{debarghya.g, ad, ajay.adsul, aparna}@csa.iisc.ernet.in

## Abstract

*Motivated by multi-distribution divergences, which originate in information theory, we propose a notion of ‘multi-point’ kernels, and study their applications. We study a class of kernels based on Jensen type divergences and show that these can be extended to measure similarity among multiple points. We study tensor flattening methods and develop a multi-point (kernel) spectral clustering (MSC) method. We further emphasize on a special case of the proposed kernels, which is a multi-point extension of the linear (dot-product) kernel and show the existence of cubic time tensor flattening algorithm in this case. Finally, we illustrate the usefulness of our contributions using standard data sets and image segmentation tasks.*

## 1. Introduction

Divergences, though introduced at the birth of information theory, Jensen-Shannon (JS) divergence appeared in the literature relatively recently [10], and the unique characteristic of this divergence is that one can measure a divergence between more than two probability distributions. Hence, one can term this as a multi-distribution divergence.

Recently, there have been a growing interest in kernel connections of the JS divergence in the machine learning community that started from the works of Endres and Schindelin [5], who observed that  $\sqrt{JS}$  is a Hilbertian metric. This renewed interests in viewing Jensen-type divergences as dissimilarity measures. Studies by Martins et al. [12] extend the idea further to the *nonextensive* case to formulate the so-called Jensen-Tsallis (JT) kernels on finite measures, that has proved to be quite useful in text classification [12] and shape recognition [3].

Though the JT-kernels and their applications have been well studied, two significant implications of these kernels have not been explored yet. The first one lies in the simple observation that JT-kernel retrieves the linear/dot-product kernel ( $x^T y$ ) in a special case, and hence, these kernels may

have interesting properties even on the Euclidian space. The second is multi-distribution nature of JS-divergence. This fact easily extends to Jensen-type kernels, in particular the JT-kernel, which leads to the notion of multi-point kernels studied in this paper.

The concept of multi-point similarities can be traced back to the studies on  $n$ -metrics, which started during works of Hayashi [8]. But its applications to unsupervised learning has been observed relatively recently [7, 1]. The works of Govindu [7] and Chen and Lerman [4] is worth mentioning in this respect, who combined multi-point similarities to spectral clustering in context of computer vision. However, the proposed methods are model specific, and hence, are restricted to applications like hybrid linear modeling and motion segmentation. On the other hand, spectral methods [14, 13] are quite general and their scope is broad ranging from image segmentation [14] to analysis of correlated mutations in HIV-1 protease [11]. To cater to the widely varying application of spectral clustering and to improve upon the Gaussian distance measure commonly employed in spectral based learning, it is quite tempting to study the spectral clustering using multi-point kernels as done in this work. Our approach distinctly deviates from the existing multi-point spectral methods in the use of multi-point kernel not restricted to model dependent similarities as in spectral curvature clustering (SCC) [4]. The contributions in this paper are listed below:

- (1) We extend the JT-kernels on finite measures to define similar kernels on the  $d$ -dimensional unit cube  $[0, 1]^d$ , that encompass the linear (dot-product) kernel. Further, we use the idea of multi-distribution divergences to define multi-point extensions of above kernels.
- (2) We develop a model-independent spectral clustering algorithm using multi-point kernels, which we call as MSC. Though MSC has an exponential complexity, like SCC [4], we prove that a cubic time complexity can be achieved in the special case of multi-point extension of linear kernel.
- (3) We study the performance of the proposed method and the kernels in the context of image segmentation.

## 2. Nonextensive Jensen-type kernels

Before going into the main discussions of this paper, we briefly review the JT-kernels on probability measures [12]. It suffices to study kernels on the  $d$ -dimensional probability simplex, denoted by  $\Delta^{d-1} = \{(p(1), \dots, p(d)) : p(j) \in [0, 1] \forall j, \sum_{j=1}^d p(j) = 1\}$ . The Jensen-Tsallis  $q$ -difference among  $n$  p.m.f.s  $p_i = (p_i(1), \dots, p_i(d)) \in \Delta^{d-1}$ ,  $i = 1, \dots, n$  is defined as [12, Section 5]

$$T_q(p_1, \dots, p_n) = H_q(\bar{p}) - \frac{1}{n^q} \sum_{i=1}^n H_q(p_i), \quad (1)$$

where  $\bar{p} = (\bar{p}(1), \dots, \bar{p}(d))$  is the p.m.f. defined as  $\bar{p}(i) = \frac{1}{n} \sum_{j=1}^n p_j(i)$ ,  $i = 1, \dots, d$ , and  $H_q$  is the nonextensive or Tsallis entropy [15] that has been extensively used in statistical mechanics to study multifractal concepts. It is given by  $H_q(p) = \frac{1}{(q-1)} (1 - \sum_{j=1}^d p(j)^q)$ , where  $q \in \mathbb{R}$ ,  $q \neq 1$  is a parameter related to the nature of the physical system. As  $q \rightarrow 1$ , the classical case of Shannon entropy is retrieved,  $H_1(p) = -\sum_{j=1}^d p(j) \ln(p(j))$ , and the  $q$ -difference (1) in this case corresponds to the JS-divergence.

Martins et al. [12] showed the theoretical justifications behind the definition of JT  $q$ -difference, and observed that  $T_q(p_1, p_2) \leq \ln_q(2)$  for all  $p_1, p_2 \in \Delta^{d-1}$ . Based on this, a kernel on probability measures  $\tilde{k}_q : \Delta^{d-1} \times \Delta^{d-1} \mapsto [0, \infty)$  was proposed as [12, Definition 26]

$$\begin{aligned} \tilde{k}_q(p_1, p_2) &= 2^q (\ln_q(2) - T_q(p_1, p_2)) \\ &= \frac{1}{(q-1)} \sum_{j=1}^d ((p_1(j) + p_2(j))^q - p_1(j)^q - p_2(j)^q) \end{aligned} \quad (2)$$

for  $q \neq 1$ , which is the Jensen-Tsallis (JT) kernel between the two probability measures  $p_1$  and  $p_2$ . The above class of kernels  $\tilde{k}_q$  is positive definite on  $\Delta^{d-1}$  for  $0 \leq q \leq 2$  [12]. For  $q = 2$ , we have a dot-product kernel on  $\Delta^{d-1}$

$$\tilde{k}_2(p_1, p_2) = 2 \sum_{j=1}^d p_1(j) p_2(j) = 2 (p_1(j))^T (p_2(j)), \quad (3)$$

and in the limit of  $q \rightarrow 1$ , we have the JS-kernel defined as

$$\begin{aligned} \tilde{k}_1(p_1, p_2) &= \sum_{j=1}^d ((p_1(j) + p_2(j)) \ln(p_1(j) + p_2(j)) \\ &\quad - p_1(j) \ln(p_1(j)) - p_2(j) \ln(p_2(j))). \end{aligned} \quad (4)$$

## 3. The notion of multi-point kernels

We now present the main idea of this paper – multi-point extensions of the JT-kernels, (2)-(4). To extend the scope of these kernels, we first extend them to the real space. More specifically, we present extensions to the set  $[0, 1]^d$ . This is

a technical requirement, and is not restrictive since it is a common practice to normalize features of data and such a set suffices for most datasets. We proceed along the lines of the defined probability kernel (2), and define an extension of JT-kernel  $k_q : [0, 1]^d \times [0, 1]^d \mapsto [0, \infty)$  of the form

$$k_q(x, y) = \begin{cases} \frac{1}{(q-1)} \sum_{j=1}^d ((x(j) + y(j))^q - x(j)^q - y(j)^q) & \text{for } q \neq 1 \\ \sum_{j=1}^d ((x(j) + y(j)) \ln(x(j) + y(j)) - x(j) \ln(x(j)) - y(j) \ln(y(j))) & \text{for } q = 1, \end{cases} \quad (5)$$

where  $x = (x(1), \dots, x(d))$ ,  $y = (y(1), \dots, y(d)) \in [0, 1]^d$ . The special case of linear kernel on  $[0, 1]^d$  follows similar to (3). The significance of JT-kernel is the fact that while the Gaussian kernel follows the nature of the Euclidean distance, similar to the linear kernel, the distance in case of Jensen-type kernels usually exhibit a skewed behavior. Further, localization effects are much less in the Jensen-type kernels as compared to the Gaussian kernel, since they are not exponentially decaying. The following result shows that the above extension does not affect the positive definiteness of the kernel. This can be proved by mimicking the proof of [12, Proposition 27] using the above kernel function.

**Proposition 1.** *JT-kernels  $k_q$  are positive definite on  $[0, 1]^d$  for all dimensions  $d$  and all  $q \in [0, 2]$ .*

We now present the multi-point extensions of the JT-kernel (5). The idea is based on the multi-distribution definition of Jensen-Tsallis  $q$ -difference (1) where  $n$  need not be equal to 2. We extend the JT-kernel for arbitrary number of points in  $\mathcal{X} = [0, 1]^d$  to obtain a class of multi-point kernels  $\{K_{q,n}\}_{n \in \mathbb{N}}$  with  $K_{q,n} : \mathcal{X}^n \mapsto [0, \infty)$  defined as

$$K_{q,n}(x_1, \dots, x_n) = \begin{cases} \frac{1}{(q-1)} \sum_{j=1}^d \left[ \left( \sum_{i=1}^n x_i(j) \right)^q - \sum_{i=1}^n (x_i(j))^q \right] & \text{for } q \neq 1 \\ \sum_{j=1}^d \left[ \left( \sum_{i=1}^n x_i(j) \right) \ln \left( \sum_{i=1}^n x_i(j) \right) - \sum_{i=1}^n x_i(j) \ln x_i(j) \right] & \text{for } q = 1. \end{cases} \quad (6)$$

The above definition is consistent with the multi-distribution extensions of JT  $q$ -difference. Since it naturally extends a positive definite kernel, we refer to it as a kernel. In the linear case, i.e., for  $q = 2$ , we retrieve a multi-point

version of the dot-product kernel as

$$K_{2,n}(x_1, \dots, x_n) = 2 \sum_{i=1}^n \sum_{j=i+1}^n x_i^T x_j, \quad (7)$$

which will be discussed in greater detail in sequel. The above extension of two-point kernels captures information about similarity among multiple points, and is capable of providing a more global measure of similarity. Further, the proposed multi-point similarity is not dependent on any geometric model, unlike the ones in [7, 4], and hence, it is applicable in a more general framework. Next, we present a spectral clustering method based on multi-point kernels. The basic approach is similar to the spectral curvature clustering (SCC) [4], but it is applicable for any multi-point kernel.

## 4. Multi-point spectral clustering

### 4.1. Algorithm

We consider the problem of clustering  $N$  points,  $\{x_1, \dots, x_N\} \in \mathcal{X}$ , into  $m$  clusters,  $C_1, \dots, C_m$ , using any  $n$ -point similarity measure  $K : \mathcal{X}^n \mapsto \mathbb{R}$ . The similarity among different points is represented by a  $n^{\text{th}}$  order  $N$ -dimensional real tensor  $\mathcal{A}$ , where  $\mathcal{A}_{i_1, i_2, \dots, i_n} = K(x_{i_1}, x_{i_2}, \dots, x_{i_n})$  for  $i_j = 1, \dots, N$  with  $j = 1, \dots, n$ . We observe from (6) that  $K$  is permutation invariant, i.e., the similarity does not change if the arguments are re-ordered. Hence, the tensor  $\mathcal{A}$  is super-symmetric. The idea is to construct a similarity (or, affinity) matrix from  $\mathcal{A}$ . This is done by tensor unfolding or mode-1 matricization [9], where we construct a matrix  $A \in \mathbb{R}^{N \times N^{n-1}}$  whose  $j^{\text{th}}$  column, for  $j = 1 + \sum_{l=2}^n (i_l - 1)N^{l-1}$  is the stack of tensor  $\mathcal{A}$  obtained by varying the first index, and fixing others at  $(i_2, \dots, i_n)$ . From  $A$ , the affinity matrix is constructed as  $V = AA^T$  that preserves the left eigenvectors of  $A$  (or mode-1 eigenvectors of  $\mathcal{A}$ ). Below, we state the algorithm based on spectral clustering algorithm due to Ng et al. [13].

The complexity of MSC is quite large since computation of each element in  $V$  requires  $2N^{n-1}$  kernel computations,

---

#### Algorithm 1 Multi-point Spectral Clustering (MSC)

---

**Given:**  $n^{\text{th}}$  order tensor  $\mathcal{A}$  representing affinity among data points  $\{x_1, \dots, x_N\} \in \mathcal{X}$ .

1. Unfold  $\mathcal{A}$  to obtain flattened matrix  $A$ , and let  $V = AA^T$ .
  2. Normalize affinity matrix as  $Z = D^{-1/2}VD^{-1/2}$ , where  $D$  is a diagonal matrix with  $d_{ii} = \sum_{j=1}^N V_{ij}$ .
  3. Compute  $u_1, \dots, u_m$ , top- $m$  unit eigenvectors of  $Z$ .
  4. Normalize rows of  $U = [u_1, \dots, u_m]$  to have unit length.
  5. Cluster the rows of  $U$  into  $m$  clusters using  $k$ -means, and partition  $\{x_1, \dots, x_N\}$  accordingly.
- 

and hence, complexity of determining  $V$  turns out to be  $O(N^{n+1})$ . We can incorporate the heuristic approach mentioned in [7], to approximate  $V$  as  $V \approx \sum_{k=1}^c w_{j_k} w_{j_k}^T$ , by uniformly sampling  $c$  columns from all the  $N^{n-1}$  columns of  $A$ , where  $w_{j_k}$  denotes the  $j_k^{\text{th}}$  column of  $A$ . Though computation reduces to a great extent to  $O(cN^2)$  for  $c \ll N^{n-1}$ , the performance of the algorithm is quite poor in general, when model underlying the data is not known a priori. In fact, since MSC does not assume geometric structures, the effect of such approximations is quite severe in this case. More efficient methods discussed in [4] in context of SCC can be used. We do not discuss such approximations here, but focus on a special case of multi-point JT-kernel, where cubic time complexity is achieved for MSC.

### 4.2. MSC using multi-point linear kernel

Recall that the multi-point JT-kernel for  $q = 2$  (7), which is a multi-point extension of linear kernel. The structure of this multi-point linear kernel helps to compute the affinity matrix  $V$  explicitly in cubic time as shown below.

**Proposition 2.** Let  $X = (x_1, x_2, \dots, x_N) \in [0, 1]^{d \times N}$  represent the given data matrix and  $\bar{x} := \sum_{i=1}^N x_i$  be the component-wise addition of the vectors. Then, the affinity matrix  $V$  corresponding to the  $n$ -point linear kernel  $K_{2,n}$  (7) can be written as

$$\begin{aligned} V = & 4 \binom{n-1}{1} N^{n-2} (X^T X)^2 + 8 \binom{n-1}{2} N^{n-3} (X^T \bar{x} \bar{x}^T X) \\ & + 8 \binom{n-1}{2} N^{n-3} (X^T X X^T \bar{x} \mathbf{1}_{1 \times N} + \mathbf{1}_{N \times 1} \bar{x}^T X X^T X) \\ & + 12 \binom{n-1}{3} N^{n-4} \|\bar{x}\|_2^2 (X^T \bar{x} \mathbf{1}_{1 \times N} + \mathbf{1}_{N \times 1} \bar{x}^T X) \\ & + 4 \binom{n-1}{2} N^{n-5} \left( N^2 \|X^T X\|_F^2 + 2(n-3)N \|X^T \bar{x}\|_2^2 \right. \\ & \quad \left. + 2 \binom{n-3}{2} \|\bar{x}\|_2^4 \right) \mathbf{1}_{N \times N} \end{aligned} \quad (8)$$

where  $\mathbf{1}_{r \times s}$  denotes a  $r \times s$  matrix of all 1's,  $\|\cdot\|_F$  is the Frobenius norm.

*Proof.* We provide a brief sketch of the proof. Note that  $V = AA^T$  and it can be written as

$$\begin{aligned} V = & 4 \sum_{i_2, \dots, i_n=1}^N \left[ X^T \left( \sum_{l=2}^n \sum_{r=2}^n x_{i_l} x_{i_r}^T \right) X \right. \\ & + X^T \left( \sum_{l=2}^n \sum_{r=2}^n \sum_{s=r+1}^n x_{i_l} x_{i_r}^T x_{i_s} \right) \mathbf{1}_{1 \times N} \\ & + \mathbf{1}_{1 \times N} \left( \sum_{r=2}^n \sum_{l=2}^n \sum_{k=l+1}^n x_{i_r} x_{i_l}^T x_{i_k} \right) X \\ & \left. + \left( \sum_{l=2}^n \sum_{r=2}^n \sum_{k=l+1}^n \sum_{s=r+1}^n x_{i_l}^T x_{i_s} x_{i_r}^T x_{i_s} \right) \mathbf{1}_{N \times N} \right], \end{aligned} \quad (9)$$

where we use the fact that given  $i_2, \dots, i_n$ , the  $j^{\text{th}}$  column of  $A$ , where  $j = (1 + \sum_{l=2}^n (i_l - 1)N^{l-2})$ , is simply

$2X^T (\sum_{l=2}^n x_{i_l}) + 2 (\sum_{l=2}^n \sum_{k=l+1}^n x_{i_l}^T x_{i_k}) \mathbf{1}_{N \times 1}$ . Comparing (9) and (8), we observe that the first term in (9) decomposes into the first two terms of (8). The second and third terms of (9) contribute to the third and fourth terms of (8), while the last term of (9) is equal to the last term in (8). Also, the outer summation in (9) may be pushed inside to simplify the results of the inner summations as shown below. For the first term, we consider the outer product of same and distinct vectors separately as

$$\begin{aligned} & \sum_{i_2, \dots, i_n=1}^N \sum_{l=2}^n \sum_{r=2}^n x_{i_l} x_{i_r}^T \\ &= N^{n-2} \sum_{l=2}^n \sum_{i_l=1}^N x_{i_l} x_{i_l}^T + N^{n-3} \sum_{r,l=2, r \neq l}^n \sum_{i_l, i_r=1}^N x_{i_l} x_{i_r}^T \end{aligned} \quad (10)$$

since the terms act as constants while summing over all indices other than  $i_l$  and  $i_r$ , and each such summation adds up  $N$  similar terms, leading to the constants outside the summations. Now, one can verify that  $XX^T = \sum_{i=1}^N x_i x_i^T$  and  $\bar{x}\bar{x}^T = \sum_{i,j=1}^N x_i x_j^T$ . Plugging this in (10), and noting that there are  $(n-1)$  terms in the first summation and  $2\binom{n-1}{2}$  terms in the second leads to the first two terms of (8). To deal with the second term of (9), it is enough to show that

$$\begin{aligned} & \sum_{i_2, \dots, i_n=1}^N \sum_{l=2}^n \sum_{r=2}^n \sum_{s=r+1}^n x_{i_l} x_{i_r}^T x_{i_s} \\ &= 2\binom{n-1}{2} N^{n-3} XX^T \bar{x} + 3\binom{n-1}{3} N^{n-4} \|\bar{x}\|_2^2 \bar{x}. \end{aligned} \quad (11)$$

The constants  $N^{n-3}$  and  $N^{n-4}$  appear as before due to summation over indices, which are absent from the terms involved. We consider the cases  $r = l$  and  $r \neq l$  separately. For  $r = l$ , we obtain half of the first term in (11) since  $\sum_{r=2}^n \sum_{s=r+1}^n \sum_{i_r, i_s=1}^N x_{i_r} x_{i_r}^T x_{i_s} = \binom{n-1}{2} XX^T \bar{x}$ . For  $r \neq l$ , the situation becomes complicated as we may have  $s = l$ . But this happens only in  $\binom{n-1}{2}$  cases, which adds up to give the remaining half of the first term in (11). The rest of the terms on the left in (11) have distinct indices, and hence, summing over them gives a term of the form  $\sum_{i,j,k=1}^N x_i x_j^T x_k = \|\bar{x}\|_2^2 \bar{x}$ . But, there are  $3\binom{n-1}{3}$  such terms, and hence, the result. Similarly, computing the other terms in (9), one can derive the expression in (8).  $\square$

The key fact in above result is that all computations in (8) are at most  $O(N^3)$ , which implies that  $V$  is computable in cubic time. Further, though the above result holds for any  $n \in \mathbb{N}$ , few simplifications are possible for  $n \leq 4$ . For instance, if  $n = 2$ , all terms vanish except first, giving  $V = 4(X^T X)^2$ , which has the same eigen structure as  $X^T X$ . Hence, spectral clustering with  $V$  is equivalent to the case of constructing affinity using the Gram matrix. We illustrate the behavior of the multi-point linear kernel with a simple example of two concentric arcs in  $[0, 1]^2$  (Figure 1).

This is an example where  $k$ -means algorithm fails. We use both Gaussian spectral clustering and MSC with  $n$ -point linear kernel, and observe that for small  $n$  (MSC) and large  $\sigma$  (Gaussian) both methods are quite similar to  $k$ -means. Accurate clustering can be achieved for Gaussian, but this requires proper tuning of  $\sigma$  as we see that even for small variations of  $\sigma$ -values considered, the results vary considerably. On the other hand, if the large number of points are considered, MSC gives accurate results. In fact, in this example, we observed that results improved with increase in  $n$ , and for  $n \geq 7$  correct clustering were always achieved.

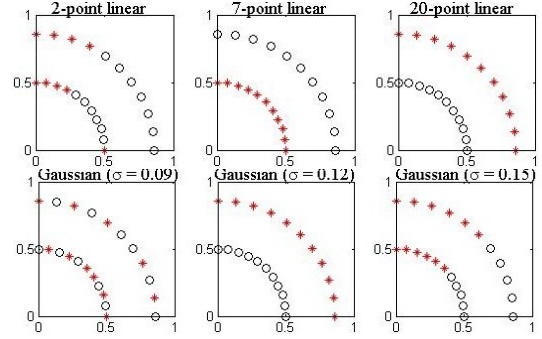


Figure 1. (top row). Clustering obtained using MSC with  $n$ -point linear kernels for  $n = 2, 7, 20$ , and (bottom row) results for Gaussian spectral clustering with  $\sigma = 0.09, 0.12, 0.15$ , respectively.

## 5. Experimental results







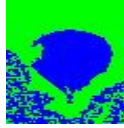






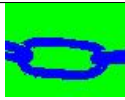
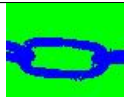
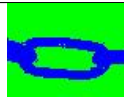


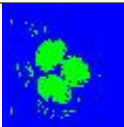
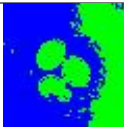

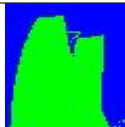















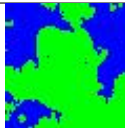
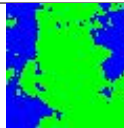
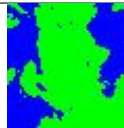

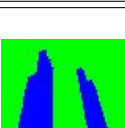
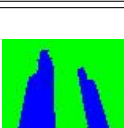
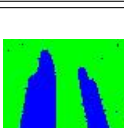
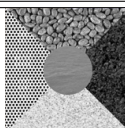
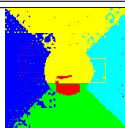
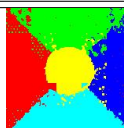
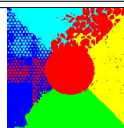
We compare the performance of MSC using the proposed multi-point JT-kernels with Gaussian spectral clustering. We do not compare our approach with methods proposed in [4, 7] since these methods require prior knowledge of geometric structures, and cannot be applied to arbitrary data sets. We perform preliminary study on standard data sets from [6]. Table 1 shows the accuracy of spectral clustering with Gaussian and 2-point JT-kernels, as well as MSC with 3-point JT-kernel and  $n$ -linear kernel. The performance measure considered is the purity of clusters obtained. For 3-point JT-kernel, the  $O(N^{n+1})$  unfolding method is used since approximations give poor performance. For JT-kernels,  $q$  is varied over  $[0, 2]$  in steps of 0.25, where JS-kernel is used for the case  $q = 1$ . We also consider MSC with the  $n$ -point linear kernel, where we vary  $n$  from 2 to 12 in steps of 2. Similarly, for Gaussian case, we tune  $\sigma$  to improve performance. We note here that tuning  $\sigma$  properly appeared to be more difficult than the parameters of our algorithms. The accuracy is averaged over multiple runs of  $k$ -means step. In Table 1, we present the best accuracy results achieved with each method. In general using 3-point JT-kernel is more effective, though computationally extensive, which indicates that considering multiple points can improve performance. The  $n$ -point linear kernels, which have reduced computational complexity also perform quite



Table 1. Comparison of MSC and Gaussian spectral clustering (for Isolet dataset, we consider only classes A,B,C).

Dataset	Gaussian SC ( $\sigma$ )	2-point JT kernel ( $q$ )	3-point JT kernel ( $q$ )	$n$ -point linear kernel ( $n$ )
Breast Cancer	0.968 (0.5)	0.963 (2.00)	<b>0.971</b> (1.0)	0.966 (6-12)
Isolet (ABC)	0.863 (10.0)	<b>0.965</b> (1.25)	<b>0.965</b> (1.0-1.25)	0.929 (4)
Iris	0.930 (0.15)	0.860 (0.0-0.5)	<b>0.965</b> (0.5)	0.792 (10)
Mammographic mass	0.799 (0.3)	0.807 (2.0)	0.776 (1.5)	<b>0.810</b> (4-12)
Semeion hand-written	<b>0.604</b> (5.0)	0.534 (1.25)	0.569 (0.25)	0.561 (4)

Figure 2. Segmentation of images using spectral clustering with Gaussian and JT-kernels, and MSC with  $n$ -point linear kernel. Each row shows results for one image, and the best parameter value for each similarity is indicated ( $q = 1$  for JT denotes the JS-kernel).

Original image	Gaussian kernel	JT-kernel	$n$ -point linear	Original image	Gaussian kernel	JT-kernel	$n$ -point linear
							
baby	$\sigma = 0.02$	$q = 1.25$	$n = 8$	balloon	$\sigma = 2.0$	$q = 1.0$	$n = 6$
							
duck	$\sigma = 0.2$	$q = 1.25$	$n = 6$	chain	$\sigma = 0.02$	$q = 1.0$	$n = 10$
							
eggs	$\sigma = 0.02$	$q = 0.5$	$n = 8$	building	$\sigma = 0.2$	$q = 1.25$	$n = 8$
							
number	$\sigma = 0.02$	$q = 0.75$	$n = 6$	leaf	$\sigma = 0.2$	$q = 0.5$	$n = 6$
							
smiley	$\sigma = 2.0$	$q = 1.0$	$n = 12$	flowers	$\sigma = 2.0$	$q = 1.25$	$n = 10$
							
tower	$\sigma = 0.2$	$q = 1.25$	$n = 8$	texture	$\sigma = 2$	$q = 1.25$	$n = 12$

well.

We use 2-point JT-kernel and  $n$ -point linear kernel for segmentation of a number of images from [2] and other sources (shown in Figure 2), where each image is reduced to  $60 \times 60$  or  $80 \times 60$ . We incorporate MSC and the proposed kernels into the segmentation approach proposed in [14]. For this, we apply JT and  $n$ -point linear kernels on the pixel intensities (in *texture* image, the intensities of the pixel and its neighbouring 8 pixels is quantized into a

histogram of 16 bins) and compute the matrix  $V$  using (8). To make our method compatible with the partitioning algorithm, spectral clustering is performed using the affinity matrix  $M = V^{(1-\lambda)} R^\lambda$ , where  $\lambda = 0.008$  and  $R$  represents the similarity matrix for pixel locations computed as below. If  $p_i$  is location of the  $i^{th}$  pixel, then  $R_{i,j} = e^{-\|p_i - p_j\|^2}$  if  $\|p_i - p_j\| < r$ , and zero otherwise. The idea is to partition the pixels into a number of clusters ( $m = 10$ ), and then group neighboring clusters till we have desired segments,

such that  $N_{cut}$  of the graph is minimized at each iteration. Figure 2 shows the best results for each of Gaussian, JT and  $n$ -point linear kernels after tuning parameters. On an average, relative times of JT and  $n$ -point linear were 0.99 and 1.03, respectively, compared to Gaussian. We observe that:

- (1) JT-kernel with  $q$  close to 1 (0.75–1.25) gives best results among all values of  $q$  in most cases.
- (2) Though 2-point linear kernel gives poor results, as  $n$  increases better partitions are obtained and mostly linear kernel over  $n = 6$  to 10 points captures all necessary details.
- (3) On the whole, Figure 2 makes it evident that in most cases, JT and  $n$ -point kernel perform at par with Gaussian similarity, and in fact, in some cases, JT-kernel shows significant improvements (for instance, *texture*, *duck*, *eggs*, *building*). The  $n$ -point linear kernel has a very simple structure, that of the dot-product, and hence, is relatively poor. However, there are instances where it still outperforms Gaussian (*balloon*, *texture*). To this end, Gaussian works significantly better than others only in *smiley* image, while segments in *flowers* for all kernels are quite different.
- (4) In one case (*flowers*), the segments obtained with JT and  $n$ -point kernels are same, but this is different from that of Gaussian. This can be justified by the similar nature of JT to  $n$ -point (extension of JT with  $q = 2$ ), that is significantly different from Gaussian.

One can note that similarities were constructed only using the pixel intensities and locations as considered in [14]. One can easily incorporate more sophisticated features into this setting, and easily use JT or  $n$ -point linear kernels to evaluate their similarities. However, by construction and justifications given in Section 3, the JT-kernel appears to be more applicable for histogram type of data such as pixel intensities. On the other hand, Gaussian similarity is more applicable for pixel distances as used here.

## 6. Discussions and concluding remarks

We develop a spectral clustering (MSC) technique that uses a similarity among more than two points. Our method is more general than existing algorithms of similar nature [7, 4], as the algorithm does not depend on the similarity measure considered. Though not discussed here, but one can easily incorporate out-of-sample extensions such as Nyström’s approximation to MSC. To extend the idea further, it would be interesting to see if spectral methods can be used on the similarity tensor, without unfolding it.

We also introduced the notion of multi-point kernels and proposed a class of multi-point similarity measures that arise out of extension of positive definite two-point kernels. We also derived a multi-point extension of linear kernels, obtained for  $q = 2$  in multi-point JT-kernel that significantly simplifies computation of MSC algorithm. Toy examples similar to Figure 1 were studied, which revealed that  $n$ -point linear kernels were always able cluster accu-

rately (above some  $n$ ) when the clusters are linearly separable. This promises a new direction of study: linear separability in the spectral clustering framework.

## Acknowledgement

D. Ghoshdastidar is supported by Google India Ph.D. Fellowship in Statistical Learning Theory.

## References

- [1] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 838–845, 2005. 1
- [2] Berkeley Segmentation Dataset and Benchmark. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>, University of California, Berkeley. 5
- [3] M. Bicego, A. F. T. Martins, V. Murino, P. M. Q. Aguiar, and M. A. T. Figueiredo. 2D shape recognition using information theoretic kernels. In *IEEE International Conference on Pattern Recognition (ICPR)*, pages 25–28, 2010. 1
- [4] G. Chen and G. Lerman. Spectral curvature clustering. *International Journal of Computer Vision*, 81(3):317–330, 2009. 1, 3, 4, 6
- [5] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003. 1
- [6] A. Frank and A. Asuncion. *UCI Machine Learning Repository*. <http://archive.ics.uci.edu/ml>, University of California, Irvine, 2010. 4
- [7] V. M. Govindu. A tensor decomposition for geometric grouping and segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1150–1157, 2005. 1, 3, 4, 6
- [8] C. Hayashi. Two dimensional quantification based on the measure of dissimilarity among three elements. *Annals of the Institute of Statistical Mathematics*, 24(1):251–257, 1972. 1
- [9] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. 3
- [10] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Info. Theory*, 37:145–151, 1991. 1
- [11] Y. Liu, E. Eyal, and I. Bahar. Analysis of correlated mutations in HIV-1 protease using spectral clustering. *Bioinformatics*, 24(10):1243–1250, 2008. 1
- [12] A. F. T. Martins, N. A. Smith, E. P. Xing, P. M. Q. Aguiar, and M. A. T. Figueiredo. Nonextensive information theoretic kernels on measures. *Journal of Machine Learning Research*, 10:935–975, 2009. 1, 2
- [13] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002. 1, 3
- [14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 1, 5, 6
- [15] C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–87, 1988. 2